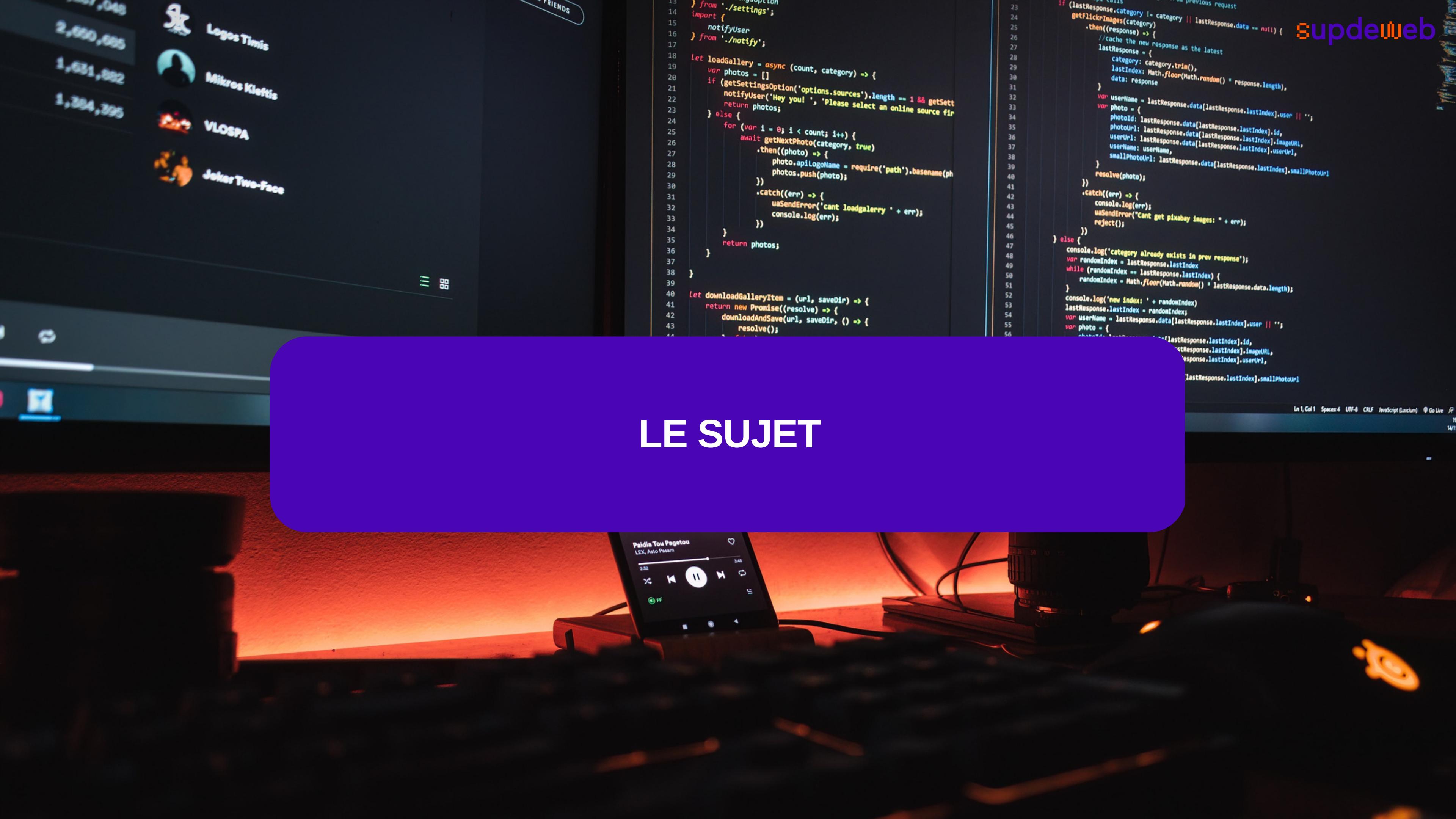




supde^Web

The Big Project



```
13 } from './settings';
14 import {
15   notifyUser
16 } from './notify';
17
18 let loadGallery = async (count, category) => {
19   var photos = []
20   if (getSettingsOption('options.sources').length == 1 && getSetting('options.sources')[0].source === 'flickr') {
21     notifyUser('Hey you!', 'Please select an online source first');
22     return photos;
23   } else {
24     for (var i = 0; i < count; i++) {
25       await getNextPhoto(category, true)
26         .then((photo) => {
27           photo.apiLogoName = require('path').basename(photo.url);
28           photos.push(photo);
29         })
30         .catch((err) => {
31           uaSendError('cant loadgalerry ' + err);
32           console.log(err);
33         });
34     }
35   }
36   return photos;
37 }
38
39 let downloadGalleryItem = (url, saveDir) => {
40   return new Promise((resolve) => {
41     downloadAndSave(url, saveDir, () => {
42       resolve();
43     });
44   });
45 }
46
47 if (lastResponse.category != category || lastResponse.data == null) {
48   getFlickrImages(category)
49     .then((response) => {
50       //cache the new response as the latest
51       lastResponse = {
52         category: category.trim(),
53         lastIndex: Math.floor(Math.random() * response.length),
54         data: response
55       };
56     });
57
58   var userName = lastResponse.data[lastResponse.lastIndex].user || '';
59   var photo = {
60     photoId: lastResponse.data[lastResponse.lastIndex].id,
61     photoUrl: lastResponse.data[lastResponse.lastIndex].imageURL,
62     userUrl: lastResponse.data[lastResponse.lastIndex].userUrl,
63     userName: userName,
64     smallPhotoUrl: lastResponse.data[lastResponse.lastIndex].smallPhotoUrl
65   };
66   resolve(photo);
67 }
68 .catch((err) => {
69   console.log(err);
70   uaSendError('Cant get pixabay images: ' + err);
71   reject();
72 })
73
74 } else {
75   console.log('category already exists in prev response');
76   var randomIndex = lastResponse.lastIndex;
77   while (randomIndex == lastResponse.lastIndex) {
78     randomIndex = Math.floor(Math.random() * lastResponse.data.length);
79   }
80   console.log('new index: ' + randomIndex);
81   lastResponse.lastIndex = randomIndex;
82   var userName = lastResponse.data[lastResponse.lastIndex].user || '';
83   var photo = {
84     photoId: lastResponse.data[randomIndex].id,
85     photoUrl: lastResponse.data[randomIndex].imageURL,
86     userUrl: lastResponse.data[randomIndex].userUrl,
87     userName: userName
88   };
89   resolve(photo);
90 }
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
747
748
749
749
750
751
752
753
754
755
756
757
757
758
759
759
760
761
762
763
764
765
766
767
767
768
769
769
770
771
772
773
774
775
776
777
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
817
818
819
819
820
821
822
823
824
825
826
827
827
828
829
829
830
831
832
833
834
835
836
837
837
838
839
839
840
841
842
843
844
845
846
846
847
848
848
849
849
850
851
852
853
854
855
856
856
857
858
858
859
859
860
861
862
863
864
865
866
866
867
868
868
869
869
870
871
872
873
874
875
876
876
877
878
878
879
879
880
881
882
883
884
885
886
886
887
888
888
889
889
890
891
892
893
894
895
895
896
897
897
898
898
899
899
900
901
902
903
904
905
905
906
907
907
908
908
909
909
910
911
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
```

LE SUJET

The Big Project

Le Pitch :

Développer un site internet offrant la possibilité d'organiser une Compétition de Jeux Vidéos rapidement.

Fonctionnalités attendues

- Une page publique présentant le produit répondant aux bonnes pratiques SEO
- Création de compte sécurisé sur la plateforme
- Création d'un événement
- Invitation de participants externes par envoie de mails
- Génération du tableau de compétitions
- Déclaration des résultats des parties dans l'application par les joueurs
- Clôture de l'événement avec reporting aux participants

Public cible : peu habitué à l'informatique

Page d'accueil

La page d'accueil intègre :

- Une image sur la page d'accueil
- Un contenu de présentation du produit
- un dispositif d'inscription / connexion

La page d'accueil est la seule page publique. Elle doit donc être optimisée pour le SEO

Création de compte

Création de compte :

- Champs obligatoires :
 - pseudo
 - adresse mail
 - choix mdp avec un champ de confirmation
- Envoi d'un mail pour confirmer la validité de l'adresse mail
- Le lien présent dans le mail permet de valider le compte

La page d'accueil est la seule page publique. Elle doit donc être optimisée pour le SEO

Création d'un événement

Création d'un événement :

- Définir le titre de la compétition
- Définir le nombre de participants
- Définir la date de début et date de fin

La page d'accueil est la seule page publique. Elle doit donc être optimisée pour le SEO

Invitation de participants externes par envoie de mails

Sur un événement donné :

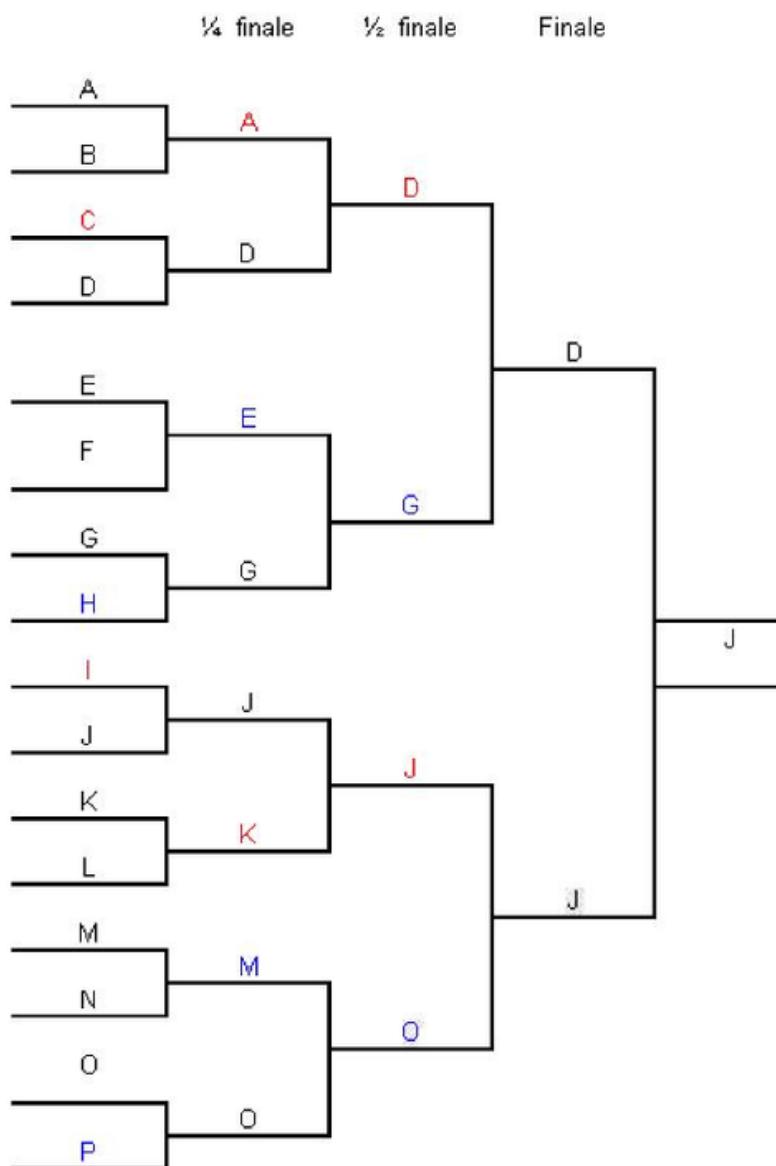
- Possibilité d'envoyer des invitations par mail sur un événement dédié
- Possibilité de partager un lien d'invitation

L'utilisateur invité devra soit créer un compte soit utilisé celui qu'il a déjà créé

La page d'accueil est la seule page publique. Elle doit donc être optimisée pour le SEO

Génération du tableau

Une fois le nombre d'inscrits invités, générer automatiquement le tableau de compétition sur le format



Déclaration d'un match

Chaque joueur doit déclarer sur son interface le résultat de son match.

Dès lors que les 2 joueurs ont saisi le même score alors le match est noté comme validé.

Le match suivant est alors affiché et ce jusqu'à la finale.

Clôture de la compétition

Lorsque la finale est validée, un mail est envoyé à l'ensemble des joueurs pour partager le tableau final de la compétition.

Contraintes

La production doit impérativement intégrer :

- Une base de donnée (relationnelle ou NoSQL)
- Un backend offrant une API
- Un frontend consommant l'API de manière sécurisée (authentification)
- L'application doit être responsive, respecter le W3C et être accessible

Choix Technologique

Backend : au choix en PHP, Python ou NodeJS (framework Symfony ou Laravel autorisé)

Frontend : HTML / CSS / Javascript (usage de ReactJS/VueJS/AngularJS autorisé)

Le frontend doit être compatible W3C, Accessible et la page publique optimisée SEO

Rappels utiles

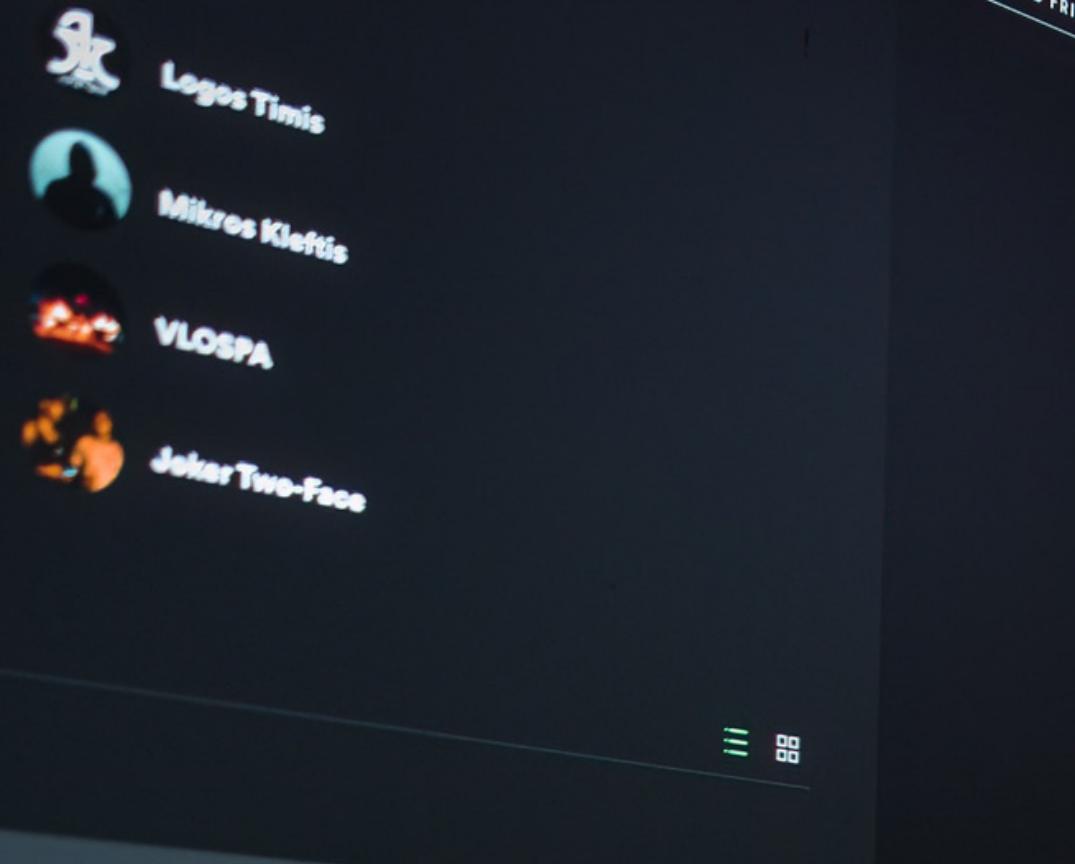
Les bases W3C : <https://openclassrooms.com/fr/courses/6691451-codez-un-site-web-accessible-avec-html-css>

Accessibilité : <https://openclassrooms.com/fr/courses/6691346-concevez-un-contenu-web-accessible>

Responsive avec bootstrap : <https://openclassrooms.com/fr/courses/7542506-creez-des-sites-web-responsives-avec-bootstrap-5>

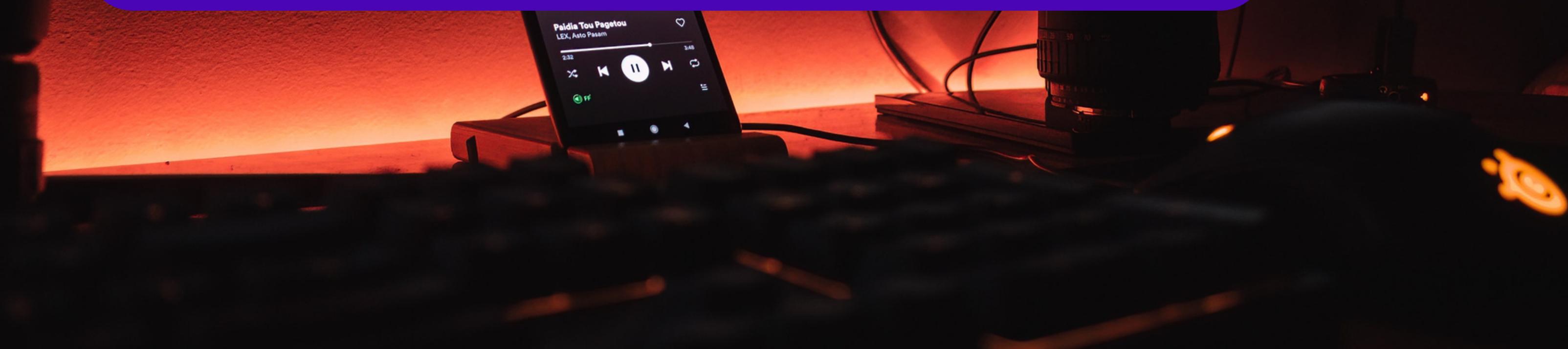
Référencement : <https://openclassrooms.com/fr/courses/5922626-optimisez-le-referencement-de-votre-site-seo-en-ameliorant-ses-performances-techniques>

Le frontend doit être compatible W3C, Accessible et la page publique optimisée SEO



```
13 } from './settings';
14 import {
15   notifyUser
16 } from './notify';
17
18 let loadGallery = async (count, category) => {
19   var photos = []
20   if (getSettingsOption('options.sources').length == 1 && getSetting('options.sources')[0] == 'flickr') {
21     notifyUser('Hey you!', 'Please select an online source first');
22     return photos;
23   } else {
24     for (var i = 0; i < count; i++) {
25       await getNextPhoto(category, true)
26         .then((photo) => {
27           photo.apiLogoName = require('path').basename(photo.url);
28           photos.push(photo);
29         })
30         .catch((err) => {
31           uaSendError('cant loadgalerry ' + err);
32           console.log(err);
33         });
34     }
35   }
36   return photos;
37 }
38
39 let downloadGalleryItem = (url, saveDir) => {
40   return new Promise((resolve) => {
41     downloadAndSave(url, saveDir, () => {
42       resolve();
43     });
44   });
45 }
46
47 if (lastResponse.category != category || lastResponse.data == null) {
48   getFlickrImages(category)
49     .then((response) => {
50       //cache the new response as the latest
51       lastResponse = {
52         category: category.trim(),
53         lastIndex: Math.floor(Math.random() * response.length),
54         data: response
55       };
56     })
57     .then((response) => {
58       var userName = lastResponse.data[lastResponse.lastIndex].user || '';
59       var photo = {
60         photoId: lastResponse.data[lastResponse.lastIndex].id,
61         photoUrl: lastResponse.data[lastResponse.lastIndex].imageURL,
62         userUrl: lastResponse.data[lastResponse.lastIndex].userUrl,
63         userName: userName,
64         smallPhotoUrl: lastResponse.data[lastResponse.lastIndex].smallPhotoUrl
65       };
66       resolve(photo);
67     })
68     .catch((err) => {
69       console.log(err);
70       uaSendError('Cant get pixabay images: ' + err);
71       reject();
72     });
73   } else {
74     console.log('category already exists in prev response');
75     var randomIndex = lastResponse.lastIndex;
76     while (randomIndex == lastResponse.lastIndex) {
77       randomIndex = Math.floor(Math.random() * lastResponse.data.length);
78     }
79     console.log('new index: ' + randomIndex);
80     lastResponse.lastIndex = randomIndex;
81     var userName = lastResponse.data[lastResponse.lastIndex].user || '';
82     var photo = {
83       photoId: lastResponse.data[randomIndex].id,
84       photoUrl: lastResponse.data[randomIndex].imageURL,
85       userUrl: lastResponse.data[randomIndex].userUrl,
86       userName: userName
87     };
88     resolve(photo);
89   }
90 }
```

LES LIVRABLES

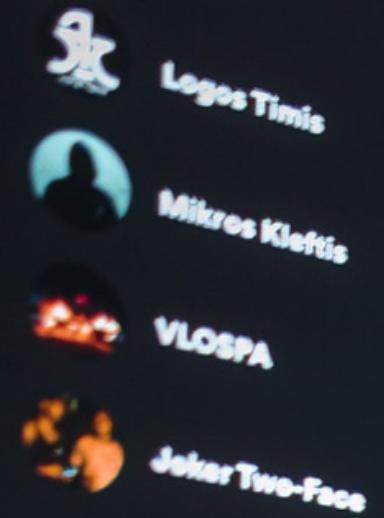


Livrable à fournir

- Code source commenté de l'application
- Modèle de la base de données
- Documentation d'installation de l'application
- Documentation fonctionnelle
- Documentation technique

Oral

- Présentation de l'architecture projet
- Argumentation sur les choix technologiques
- Démonstration

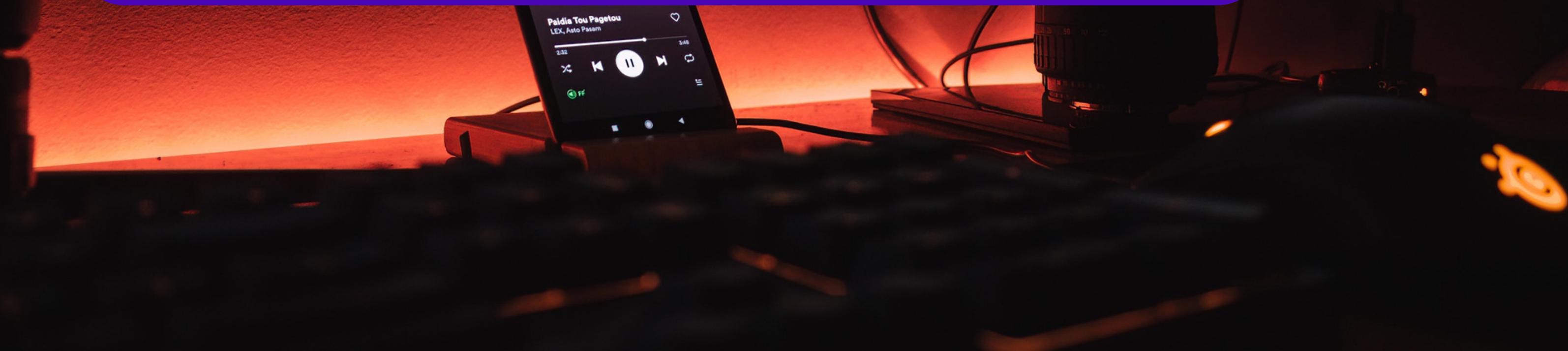


```
13 } from './settings';
14 import {
15   notifyUser
16 } from './notify';
17
18 let loadGallery = async (count, category) => {
19   var photos = []
20   if (getSettingsOption('options.sources').length == 1 && getSetting('options.sources')[0] === 'pixabay') {
21     notifyUser('Hey you!', 'Please select an online source first');
22     return photos;
23   } else {
24     for (var i = 0; i < count; i++) {
25       await getNextPhoto(category, true)
26         .then((photo) => {
27           photo.apiLogoName = require('path').basename(photo.url);
28           photos.push(photo);
29         })
30         .catch((err) => {
31           uaSendError('cant loadgalerry ' + err);
32           console.log(err);
33         })
34     }
35   }
36   return photos;
37 }
38
39 let downloadGalleryItem = (url, saveDir) => {
40   return new Promise((resolve) => {
41     downloadAndSave(url, saveDir, () => {
42       resolve();
43     });
44   });
45 }
```

```
19 // previous request
20
21 if (lastResponse.category != category || lastResponse.data == null) {
22   getFlickrImages(category)
23     .then((response) => {
24       //cache the new response as the latest
25       lastResponse = {
26         category: category.trim(),
27         lastIndex: Math.floor(Math.random() * response.length),
28         data: response
29       }
30     })
31     .catch((err) => {
32       console.log(err);
33       uaSendError('Cant get pixabay images: ' + err);
34     })
35   }
36   var userName = lastResponse.data[lastResponse.lastIndex].user || '';
37   var photo = {
38     photoId: lastResponse.data[lastResponse.lastIndex].id,
39     photoUrl: lastResponse.data[lastResponse.lastIndex].imageURL,
40     userUrl: lastResponse.data[lastResponse.lastIndex].userUrl,
41     userName: userName,
42     smallPhotoUrl: lastResponse.data[lastResponse.lastIndex].smallPhotoUrl
43   }
44   resolve(photo);
45 }
46 .catch((err) => {
47   console.log(err);
48   uaSendError('Cant get pixabay images: ' + err);
49   reject();
50 })
51 } else {
52   console.log('category already exists in prev response');
53   var randomIndex = lastResponse.lastIndex;
54   while (randomIndex == lastResponse.lastIndex) {
55     randomIndex = Math.floor(Math.random() * lastResponse.data.length);
56   }
57   console.log('new index: ' + randomIndex)
58   lastResponse.lastIndex = randomIndex;
59   var userName = lastResponse.data[lastResponse.lastIndex].user || '';
60   var photo = {
61     photoId: lastResponse.data[randomIndex].id,
62     photoUrl: lastResponse.data[randomIndex].imageURL,
63     userUrl: lastResponse.data[randomIndex].userUrl,
64     userName: userName,
65     smallPhotoUrl: lastResponse.data[randomIndex].smallPhotoUrl
66   }
67 }
```

Ln 1, Col 1 | Spaces: 4 | UTF-8 | CRLF | JavaScript (Luxium) | Go Live | 14/1

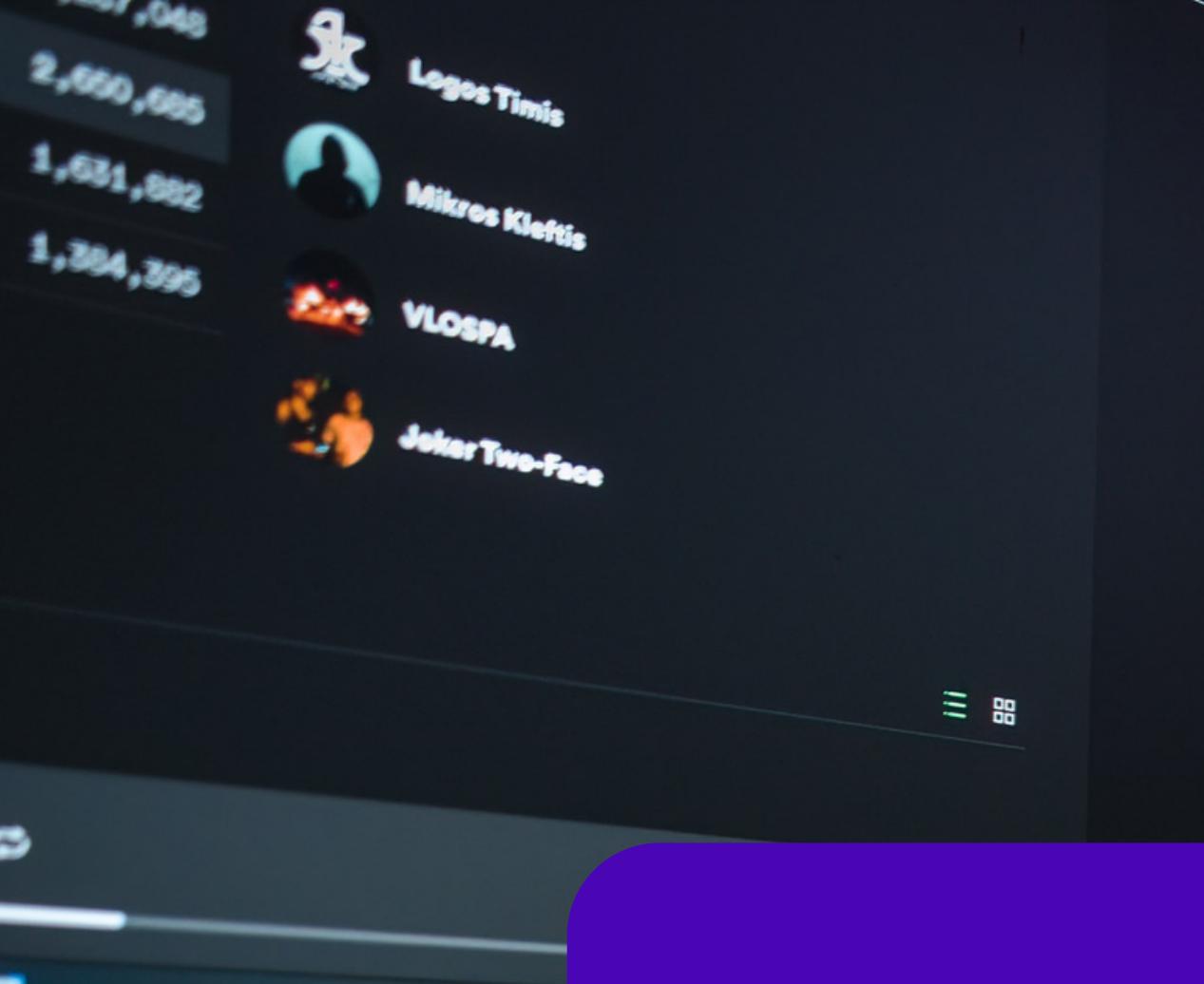
LE PLANNING DE LA SEMAINE



	Lundi 03/04/2023	Mardi 04/04/2023	Mercredi 05/04/2023	Jeudi 06/04/2023	Vendredi 07/04/2023
9h00 - 10h00					
10h00 - 11h00					
11h00 - 12h00	Florent Descroix - The Big Project	Autonomie Projet	Autonomie Projet	Maxime Blanc - Méthodologie Projet : l'agilité	Nicolas Perez - The Big Project
12h00 - 13h00		Repas			Repas
13h00 - 14h00	Repas		Repas	Repas	
14h00 - 15h00		Nicolas Perez - The Big Project			
15h00 - 16h00	Autonomie Projet		Florent Descroix - The Big Project	Florent Descroix - Javascript : Les bases, ECMAScript, NodeJS...	Florent Descroix - The Big Project
16h00 - 17h00		Autonomie Projet			



Cours



```
13 } from './settings';
14 import {
15   notifyUser
16 } from './notify';
17
18 let loadGallery = async (count, category) => {
19   var photos = []
20   if (getSettingsOption('options.sources').length == 1 && getSetting('options.sources')[0] == 'flickr') {
21     notifyUser('Hey you!', 'Please select an online source first');
22     return photos;
23   } else {
24     for (var i = 0; i < count; i++) {
25       await getNextPhoto(category, true)
26         .then((photo) => {
27           photo.apiLogoName = require('path').basename(photo.url);
28           photos.push(photo);
29         })
30         .catch((err) => {
31           uaSendError('cant loadgalerry ' + err);
32           console.log(err);
33         });
34     }
35   }
36   return photos;
37 }
38
39 let downloadGalleryItem = (url, saveDir) => {
40   return new Promise((resolve) => {
41     downloadAndSave(url, saveDir, () => {
42       resolve();
43     });
44   });
45 }
46
47
48
49
50
51
52
53
54
55
56
```

Ln 1, Col 1 | Spaces: 4 | UTF-8 | CRLF | JavaScript (Luxium) | Go Live |

BLOCS DE COMPETENCES



Evaluations

Bloc de compétences 3 - Domaine 3.1

3. Développer une solution digitale

Développement technique d'une application et d'une base de données



Travail individuel



Production écrite, orale et digitale



Projet numérique

Réalisation d'un projet numérique - Production écrite, orale et digitale

Modalité d'organisation : En analysant son cahier des charges et ses spécifications techniques et fonctionnelles, le candidat développera une solution digitale, en élaborant les composants des programmes et les bases de données garantissant un fonctionnement optimal et sécurisé de la solution.

Production évaluée : Le développement technique d'une application et d'une base de données

Seront pris en compte :

- La qualité de l'analyse des fonctionnalités attendues
- La qualité de l'analyse des spécifications techniques
- La pertinence de l'identification des contraintes techniques
- Le respect des normes et règles syntaxiques de programmation
- La clarté, logique et cohérence de la structuration du code
- La qualité et pertinence des commentaires du code
- La performance, conformité et sécurité de l'application développée

Travail individuel

Evaluations (suite)

Bloc de compétences 3 - Domaine 3.1

3. Développer une solution digitale

Critères d'évaluation

- Qualité de l'analyse des fonctionnalités attendues
- Qualité de l'analyse des spécifications techniques
- Pertinence de l'identification des contraintes techniques
- Respect des normes et règles syntaxiques de programmation
- Clarté, logique et cohérence de la structuration du code
- Qualité et pertinence des commentaires du code
- Performance, conformité et sécurité de l'application développé

Evaluations

Bloc de compétences 4 - Domaine 4.1

4. Intégrer une solution digitale dans un environnement web

Intégration de la solution digitale sur une plateforme web

 Travail individuel  Production digitale  Projet numérique

Réalisation d'un projet numérique - Production digitale

Modalité d'organisation : en tenant compte des contraintes d'accessibilité et de la pluralité des supports de consultation de l'utilisateur, le candidat intégrera la solution digitale dans un environnement web, en s'assurant de la conformité de son ergonomie et de son aspect visuel, et en mettant en œuvre les techniques permettant l'optimisation de son référencement

Production évaluée : l'intégration de la solution digitale sur une plateforme web

Seront pris en compte :

- La qualité de l'analyse des environnements utilisateurs
- La prise en compte des contraintes d'accessibilité
- La pertinence de la maquette fonctionnelle
- La qualité et la conformité de l'affichage au regard du graphisme et de l'ergonomie prévues
- La simplicité et l'intuitivité de l'interface de navigation
- Le respect des standards de la W3C
- L'effectivité de la portabilité et de l'interopérabilité de la solution
- L'optimisation du référencement naturel

Travail individuel

Evaluations (suite)

Bloc de compétences 4 - Domaine 4.1

4. Intégrer une solution digitale dans un environnement web

Critères d'évaluation

- Qualité de l'analyse des environnements utilisateurs
- Prise en compte des contraintes d'accessibilité
- Pertinence de la maquette fonctionnelle
- Qualité et conformité de l'affichage au regard du graphisme et de l'ergonomie prévues
- Simplicité et intuitivité de l'interface de navigation
- Respect des standards de la W3C
- Effectivité de la portabilité et de l'interopérabilité de la solution
- Optimisation du référencement naturel

Evaluations

Bloc de compétences 6 - Domaine 6.1

6. Assurer la mise en exploitation et la maintenance d'une solution digitale

Assurer la mise en exploitation et la maintenance d'une solution digitale

1/ Formalisation de la documentation technique/utilisateur de la solution digitale



Travail individuel



Production écrite



Projet numérique

Modalité d'organisation : en mettant en œuvre les outils et actions permettant son implémentation et favorisant son appropriation par les utilisateurs, assurer la mise en exploitation et la maintenance de la solution digitale, en veillant à sa mise à jour et son adaptation au regard de l'évolution des technologies et des besoins utilisateurs.

Production évaluée :

- la formalisation de la documentation technique/utilisateur de la solution digitale :

Seront pris en compte :

- L'exhaustivité des informations
- La clarté de la présentation
- L'utilisabilité du document
- L'adaptation du niveau de complexité au public visé

Travail individuel

Evaluations (suite)

Bloc de compétences 6 - Domaine 6.1

6. Assurer la mise en exploitation et la maintenance d'une solution digitale

Critères d'évaluation : la formalisation de la documentation technique/utilisateur de la solution digitale

- Exhaustivité des informations
- Clarté de la présentation
- Utilisabilité du document
- Adaptation du niveau de complexité au public vis